Please amend the claims in accord with the listing of claims below.

Claim 1:

1. (Currently Amended) A circuit apparatus associated with a mid-level cache for handling side-door communal software lock (CSWL) inquiries in a multiple instruction-processor computer system said computer system having other mid-level caches with similar circuit apparatae associated with said other mid-level caches, said circuit apparatus comprising:

a. inquiry generator for generating said CSWL inquiries, said CSWL inquiries being signals containing either CSWL requests or status reports regarding locked status of a CSWL, and for providing such ~~inquiries~~ CSWL requests and status reports to a side door of ~~another~~ one of said other mid-level caches ~~side door~~,

b. receiving circuit for receiving said ~~inquiries~~ CSWL requests and status reports from said similar circuit apparatae associated with said other mid-level caches,

c. Interpreter for reading the signals in ~~said~~ a received CSWL inquiry to determine if it relates to a CSWL mapped to said associated mid-level cache and what each particular lock request function requires for ~~any of~~ said received CSWL ~~requests~~inquiry,

d. CSWL cache memory within said associated mid-level cache for storing CSWLs to which said associated mid-level cache is mapped, and means for determining if a CSWL which is subject to said received CSWL inquiry is present within said CSWL cache memory,

e. comparator circuit for determining a current value of a requested CSWL within said mid-level cache's memory,

f. CSWL inquiry processor for processing said received CSWL inquiry and for generating a response to said received CSWL inquiry,

g. A circuit for responding to a requesting local processor with ~~the~~ a status received from a status report.

Claim 2:

2. (Original) The circuit apparatus of claim 1 wherein said CSWL inquiry processor is said means for determining of part "d".

Claim 3:

3. (Original) The circuit apparatus of claim 1 wherein said CSWL inquiry processor includes said comparator circuit.

Claim 4:

4. (Original) The circuit apparatus of claim 1 further comprising a CSWL map directory for said associated mid-level cache containing addresses for each CSWL to which said associated mid-level cache is mapped and wherein said CSWL inquiry processor further comprises a circuit for determining whether a CSWL inquiry is mapped to said associated mid-level cache.

Claim 5:

5. (Original) The circuit apparatus of claim 4 further comprising a lock request generator for generating an inter-mid-level cache lock requests to send to other, non-associated mid-level caches if said CSWL inquiry processor determines a CSWL inquiry is not mapped to said associated mid-level cache.

Claim 6:

6. (Original) The circuit apparatus of claim 4 further comprising a lock request generator for generating an inter-mid-level cache lock requests to send a CSWL lock function to the CSWL data determined to be mapped to another, non-associated mid-level cache if said CSWL inquiry processor determines a CSWL inquiry is not mapped to said associated mid-level cache but is mapped to said non-associated mid-level cache.

Claim 7:

7. (Original) The circuit apparatus of claim 5 wherein said lock request generator for generating lock requests determines from information in said CSWL map directory which other, non-associated mid-level cache to which to direct said inter-mid-level cache

lock request.

Claim 8:

8. (Currently Amended) The circuit apparatus of claim 1 further comprising a status stripper ~~(?)~~ circuit for fashioning a signal from a status field in a CSWL after processing by said CSWL inquiry processor to supply information needed to provide a reply to said CSWL inquiry.

Claim 9:

9. (Currently Amended) The circuit apparatus of claim 1 further comprising a side-door circuit for receiving said CSWL ~~requests~~ inquiries and status reports from other, non-associated mid-level caches ~~having similar circuit apparatus to said claim 1 circuit apparatus through which to communicate with said other, non-associated mid-level caches said similar circuit apparatuses~~.

Claim 10:

10. (Currently Amended) The circuit apparatus of claim 8 wherein CSWL ~~requests~~ inquiries from processor units associated with said associated mid-level cache are received by said circuit apparatus through an internal data channel.

Claim 11:

11. (Original) The circuit apparatus of claim 1 wherein access to said lock cache is given a lower priority than access to a data cache in said associated mid-level cache.

Claim 12:

12. (Currently Amended) A computer system having a set of mid-level caches wherein said mid-level caches are connected through ~~said~~ a side door in each of said mid-level caches to side doors ~~of said similar circuit apparatae in each of said non-associated mid-level caches~~in mid-level caches of other ones of said set of mid-level caches, and wherein a radial communications pathway joins all such side doors to enable side door communication from ones of said set of said mid-level caches to other ones of said set of mid-level caches.

Claim 13:

13. (Currently Amended) A computer system as set forth in claim ~~9~~ 12 wherein the radial is a bus and the connection to each mid-level cache from said radial is said ~~a~~ side door of each of said mid-level caches and wherein said side door is programmed to respond only to mapped CSWLs appearing on the bus.

Claim 14:

14. (Currently Amended) A computer system as set forth in claim ~~9~~ 12 wherein the radial is a crossbar and the connections are configured by mapping of said CSWLs such that a given CSWL will map to a unique mid-level cache.

Claim 15:

15. (Currently Amended) A method for handling communal software locks (CSWLs) among a set of controller circuits situated in an associated set of mid-level caches in a multiprocessor computer system wherein each controller circuit is associated to a one of said set of mid-level caches, said method, comprising:

A) receiving a request for a software lock by a one of said set of controller circuits in a receiving one of said set of controller circuits (a receiving controller) from a requester,

B) interpreting said software lock request and if said interpreting yields a determination that said software lock request relates to a CSWL, then:

C) determining if said requested CSWL is mapped to said receiving controller and if mapped to said receiving controller:

1. searching said associated mid level cache for presence of said CSWL in said associated mid level cache,
2. if said requested CSWL is in a storage circuit in said associated mid level cache either:
   a. setting said requested CSWL and returning an ownership indicia to said requester, or
   b. if said requested CSWL is owned by another, returning a status to said requester,

3. if said requested CSWL is not in a storage circuit within said associated mid level cache:

a. forming a data request for the requested CSWL and sending said data request to said multiprocessor computer system to retrieve the requested CSWL from a current owner

b. receiving said requested CSWL from said multiprocessor computer system and processing said request in accord with sub-step 2, and

D) if said software lock request does not relate to a CSWL, passing said software lock request as ordinary data within said computer system.

Claim 16:

16. (Original) The method of claim 15 wherein step C 2) is performed by said receiving controller in said associated cache and wherein said requested CSWL is retained by said associated mid level cache.

Claim 17:

17. (Original) The method of claim 15 wherein said interpreting of step B comprises; recognizing whether said request has been received into said associated mid level cache through a side door, and if received through said side door then determining that said request is a CSWL request.

Claim 18:

18. (Original) The method of claim 16 wherein said interpreting of step B comprises:

setting a flag indicator by a processor which uses said associated mid level cache to indicate that a software lock request is a CSWL request and recognizing said flag indicator by said receiving controller.

Claim 19:

19. (Original) The method of claim 15 further comprising prioritizing step C wherein step C will be performed by said set of mid level caches only after other mid level cache functions.

Claim 20:

20. (Currently Amended) The method of claim 15 further comprising prioritizing step C to be performed at a lower priority than at least one other functions of said mid level cache, on an interleaved basis wherein said at least one other function includes but is not limited to transferring data.